AMENDMENTS

In the Claims

The following is a marked-up version of the claims with the language that is underlined ("___") being added and the language that contains strikethrough ("~~___~~") being deleted:

1-44. (Cancelled)

45. (Previously Presented) A method of encrypting multi-media data flow packets, comprising the steps of:

receiving a series of multi-media data flow packets, each packet comprising a sequence number;

storing the series of multi-media data flow packets in a jitter buffer;

re-sequencing the series of multi-media data flow packets into a pseudo-random order; and

transmitting each multi-media data flow packet in the re-sequenced series in the re-sequenced order.

46. (Previously Presented) The method of claim 45, wherein said re-sequencing uses a randomization code that is algorithmically predictable if a key to said randomization code is known.

47. (Previously Presented) The method of claim 45, further comprising the step of performing bit manipulation within said first multi-media data flow packet.

48. (Previously Presented) The method of claim 47, wherein said step of performing bit manipulation is performed by using a bit-size operation that is restorable.

49. (Previously Presented) The method of claim 48, wherein said bit-size operation comprises negation.

50. (Previously Presented) The method of claim 45, further comprising the step of pseudo-randomly shuffling a destination address of each of the multi-media data flow packets.

51. (Previously Presented) The method of claim 50, wherein said destination address is a destination port address of said second endpoint.

52. (Previously Presented) A computer readable medium for encrypting multi-media data flow packets, the program for performing the steps of:

receiving a series of multi-media data flow packets;

storing the series of multi-media data flow packets in a jitter buffer;

re-sequencing the series of multi-media data flow packets into a pseudo-random order; and

transmitting each multi-media data flow packet in the re-sequenced series in the re-sequenced order.

53. (Previously Presented) The computer readable medium of claim 52, wherein said re-sequencing uses a randomization code that is algorithmically predictable if a key to said randomization code is known.

54. (Previously Presented) The computer readable medium of claim 52, the program further comprising logic for performing the step of performing bit manipulation within said first multi-media data flow packet.

55. (Previously Presented) The computer readable medium of claim 54, wherein said step of performing bit manipulation is performed by using a bit-size operation that is restorable.

56. (Previously Presented) The computer readable medium of claim 55, wherein said bit-size operation comprises negation.

57. (Previously Presented) The computer readable medium of claim 52, the program further comprising logic for performing the step of pseudo-randomly shuffling a destination address of each of the multi-media data flow packets.

58. (Previously Presented) The computer readable medium of claim 57, wherein said destination address is a destination port address of said second endpoint.

59. (Currently Amended) A system for encrypting multi-media data flow packets, comprising:

a transceiver;

software ~~stored within said first endpoint~~ defining functions to be performed by the system; and

a processor configured by said software to perform the steps of:

receiving a series of multi-media data flow packets;

storing the series of multi-media data flow packets in a jitter buffer;

re-sequencing the series of multi-media data flow packets into a pseudo-random order; and

transmitting each multi-media data flow packet in the re-sequenced series in the re-sequenced order.

60. (Previously Presented) The system of claim 59, wherein said re-sequencing uses a randomization code that is algorithmically predictable if a key to said randomization code is known.

61. (Previously Presented) The system of claim 59, processor configured by said software to perform the step of pseudo-randomly shuffling a destination address of each of the multi-media data flow packets.

62. (Previously Presented) The system of claim 61, wherein said destination address is a destination port address of said second endpoint.

63-66. (Cancelled)

67. (Currently Amended) A method of encrypting a series of multi-media data flow packets, comprising the steps of:

receiving a series of multi-media data flow packets belonging to a first flow, each packet in the series having ~~the same port address~~ a port address that is the same as the port address of the other packets in the series;

generating a pseudo-random sequence of numbers, the sequence of numbers associated with the port address;

replacing the port address in each packet with the product of ~~the~~ a corresponding number in the sequence of numbers and the size of the sequence; and

transmitting each packet to a receiver.

68. (Cancelled)

69. (Cancelled)

70. (Previously Presented) The method of claim 67, wherein the generating step uses a randomization code that is predictable if a key to the randomization code is known.

71. (Previously Presented) The method of claim 70, wherein the key is known to the receiver.

72. (Previously Presented) The method of claim 67, wherein the size of the sequence is known to the receiver.

73. (Previously Presented) The method of claim 67, wherein the port address comprises a destination port address.